

METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT FOR
SCHEDULING UPSTREAM COMMUNICATIONS

Inventors: Dolors Sala
Ajay Chandra V. Gummalla
John O. Limb
Richard Protus

CROSS-REFERENCE TO RELATED APPLICATIONS

- [0001] This application claims benefit of U.S. Patent Serial No. 09/427,792, entitled "System and Method for Multiplexing Data from Multiple Sources," filed October 27, 1999, by Limb *et al.*, (still pending), which is incorporated by reference herein in its entirety; and benefit of the following United States provisional applications:
- [0002] U.S. Patent Serial No. 60/182,470, entitled "Intelligent Silence Suppression," filed February 15, 2000, by Gummalla *et al.*, (still pending), which is incorporated by reference herein in its entirety;
- [0003] U.S. Patent Serial No. 60/247,188 (Attorney Docket No. CPH 40892 (BP 1560)), entitled "A Local Scheduling Mechanism for Cable Modems," filed November 9, 2000, by Sala *et al.*, (still pending), which is incorporated by reference herein in its entirety;
- [0004] U.S. Patent Serial No. 60/254,415 (Attorney Docket No. CPH 40892 (BP 1560.1)), entitled "A Local Scheduling Mechanism for Cable Modems," filed December 8, 2000, by Sala *et al.* (still pending), which is incorporated by reference herein in its entirety;

- [Handwritten signature]*

Sub
A1

Medium," U.S. Patent Serial No. TBD (Attorney Docket No. 1875.0450001), by Gummalla *et al.*, filed concurrently herewith and incorporated by reference herein in its entirety.

BACKGROUND OF THE INVENTION

Field of the Invention

[0012] The present invention is directed to controlling network transmissions. More particularly, the present invention relates to scheduling transmissions from multiple clients in a network environment.

Background Art

[0013] With the advent of the Internet, it has become more commonplace to develop vast communications networks to readily exchange information over remote areas. As modern technology continues to evolve to create new services to be provided over communications media, a greater demand has been generated for bandwidth and improved quality of services. For example, television broadcasts historically involved one-way communication from a broadcast transmitter to a viewer's home. As interactive or personal television services continue to grow, communications media used to support one-way communications must now contend with an increased demand for bi-directional communications.

[0014] In a conventional communications network, a communications device (such as a modem) would request bandwidth from a headend prior to transmitting data to its destination. The headend would allocate bandwidth to the cable modem based on availability and competing demands from other modems. The allocation of bandwidth is typically granted to the requesting modem in a MAP. The cable

modem would be required to follow the instructions specified in the MAP, and use the grant for the service specified in the MAP.

[0015] Problems arise when the service specified in the MAP is later determined to no longer require the bandwidth, or require more bandwidth than originally requested. Another problem can occur if another service of equal or higher priority should require immediate bandwidth shortly after the headend's granting a lower-priority service's request bandwidth. The cable modem may not be able to use the granted bandwidth to transmit data from the higher priority service, because the grant would be restricted to the lower priority service. For example, a DOCSIS-compliant network system specifies that a cable modem must accept decisions made during the requesting phase.

[0016] One mechanism that can be implemented to reduce latency would be to utilize piggyback requests for bandwidth. Piggyback requests can be very effective if a cable modem is operating in a contention mode, where the modem transmit packets without a specified grant. Transmitting a signal during a contention mode increases the likelihood of the packets colliding, getting loss or becoming corrupted.

[0017] However, the conventional way of piggybacking requests is to use variable sized headers. The header can be extended to incorporate the piggyback request when there exist a need to send one. This is a common practice in a DOCSIS-compliant environment. This approach is more effective if the header is very small and/or the size of the request message is also small. Otherwise, piggybacking requests can add excessive packet overhead that require more bandwidth or may cause packet latency.

[0018] Consequently, a system and method are needed to solve the above-identified problems and provide a simple, efficient and cost-effective way to schedule communications in an classify packets in a dynamic environment.

BRIEF SUMMARY OF THE INVENTION

- [0019] The present invention provides a system and method for scheduling transmissions from a plurality of services operating over a widely distributed communications network. In an embodiment, a headend communications device (such as a cable modem termination system) functions as an arbitrator of bandwidth among a plurality of cable modem configurable for bi-directional communications. The headend grants a bandwidth region to a specified cable modem or assigns contention regions for a group of cable modems.
- [0020] Each cable modem contains a local scheduler that sends requests for bandwidth according to local policies or rules. Upon receipt of a grant from the headend, the local scheduler selects packets to be transmitted to best serve the needs of the services associated with the cable modem. Accordingly, a service requesting bandwidth may not be the service utilizing the grant corresponding to bandwidth request. Nonetheless, the local scheduler manages bandwidth allocation among its local services such that all requesting services eventually receive bandwidth.
- [0021] In an embodiment, the cable modem contains multiple priority queues for collected packets from different services. The local scheduler is programmable to transmit packets from the priority queues based on the actual priority of the associated service. For example, local rules may specify that voice telephony services have a higher priority than video conferencing.
- [0022] In an embodiment, independent piggybacking is used to request additional bandwidth. The present invention allows single piggybacking, multiple piggybacking and cross-piggybacking.

BRIEF DESCRIPTION OF THE DRAWINGS/FIGURES

- [0023] The accompanying drawings, which are incorporated herein and form part of the specification, illustrate the present invention and, together with the description, further serve to explain the principles of the invention and to enable a person skilled in the pertinent art to make and use the invention. In the drawings, like reference numbers indicate identical or functionally similar elements. Additionally, the leftmost digit(s) of a reference number identifies the drawing in which the reference number first appears.
- [0024] FIG. 1 illustrates a data traffic management system according to an embodiment of the present invention.
- [0025] FIG. 2 illustrates an operational flow diagram for the steps involved in scheduling communications according to an embodiment of the present invention.
- [0026] FIG. 3 illustrates a block diagram of an example computer system useful for implementing the present invention.
- [0027] FIGS. 4a and 4b illustrate an operational flow diagram for the operating states of a cable modem.

DETAILED DESCRIPTION OF THE INVENTION

Table of Contents

- I. Data Traffic Management System Overview
- II. Cable Modem Scheduling
- III. Piggybacking Bandwidth Requests
- IV. Conclusion

I. Data Traffic Management System Overview

[0028] FIG. 1 illustrates data traffic management system 100 according to an embodiment of the present invention. System 100 is preferably, but not necessarily, of the type described in U.S. Patent Serial No. TBD (Attorney Docket No. CPH 40892 (BP 1560)), entitled "A Local Scheduling Mechanism for Cable Modems," filed November 9, 2000, by Sala *et al.*, (still pending), which is incorporated by reference herein in its entirety.

[0029] System 100 includes a headend or cable modem termination system (CMTS) 102 that exchanges data with one or more cable modems 104 over a communications interface 110, which includes wired or wireless local area networks (LAN) or wide area networks (WAN), such as an organization's intranet, local internets, the global-based Internet (including the World Wide Web (WWW), private enterprise networks, or the like. Communications interface 110 includes wired, wireless or both, transmission media, including satellite, terrestrial (fiber optic, copper, coaxial and the like), radio, microwave and any other form or method of transmission. In an embodiment, CMTS 102 and cable modem 104 can be integrated to support protocols, such as, Internet Protocol (IP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Real Time Transport Protocol (RTP), Resource Reservation Protocol (RSVP), or the like.

[0030] One or more downstream channels carry information (such as, television signals, IP data packets, control messages in MPEG format) from CMTS 102 to the plurality cable modems 104. Similarly, one or more upstream channels carry bursts of packets from the cable modems 104 to CMTS 102. In an embodiment, the bursts are assigned or allocated by mini-slots prior to upstream transmissions. In another embodiment, the bursts are quantified and apportioned by bits, bytes, mini-ticks (such as 2.5 μ m per click) or like metrics for apportioning bandwidth regions. The burst includes bandwidth requests from cable modems 104 and

transmissions of data from the corresponding services, as discussed in greater detail below.

[0031] CMTS 102 includes an upstream scheduler 106 that arbitrates bandwidth requirements among multiple cable modems 104. Map builder 108 also is included within CMTS 102 to transmit a MAP containing upstream slot specifications and grant specifications to cable modems 104. In an embodiment, MAPs are introduced into the MPEG transport stream of downstream communication as control messages.

[0032] Bursts from cable modems 104 are received by burst demultiplexer 122, which forms the physical layer interface between CMTS 102 and the upstream channel from communications interface 110. Burst demultiplexer 122 sends bandwidth request to request queue 124 and other packets to upper layer 126, which forwards the packet to another device or application, such as a web browser, another cable modem 104, or other data receiver. Packets from these other data receivers are collected by output queue 128 for external use.

[0033] A contention slot allocator (CSA) 130 specifies which areas in an upstream channel are to be used as assigned bandwidth regions for grants and contention transmissions. In an embodiment, if no requests are resident in request queue 124, the bandwidth region is allocated to contention transmissions. In an embodiment, a piggyback probability metric is sent from request queue 124 to CSA 130 to aid CSA 130 in determining the percentage of bandwidth to allocate for grants and contention transmissions.

[0034] The state of request queue 124 is sent to call admission controller (CAC) 132 that decides whether to admit more traffic into system 100. In an embodiment CAC 132 is operable to process voice data and generate a call admission signal if a call is accepted. The call admission signal is sent downstream to the cable modem 104 that is requesting a call admission. However CAC 132 is not limited for use with voice data. CAC 132 can support other forms of media and multimedia. CAC 132 also sends the parameters for establishing

queues within upstream scheduler 106, and upstream scheduler 106 notifies CAC 132 after the queues have been established. Upstream scheduler 106 also received information from CSA 130 representative of the distribution of contention bandwidth regions or contention mini-slots (CMS). Based on information received from CAC 132, CSA 130 and a collision resolution algorithm device (CRA) 138, upstream scheduler 106 specifies the bandwidth regions for grants and contention requests.

[0035] Collision detector 136 monitors contention regions in the upstream channels on a continuous basis to detect a collision immediately if a collision occurs. Collision detector 136 sends a collision/no collision (C/NC) signal to CRA 138, which uses the C/NC signal to adjust the CRA parameters. This allows the C/NC signal and an acknowledgment/no acknowledgment message to be sent downstream to enable cable modem 104 to resent the collided request.

[0036] Each cable modem 104 hosts one or more services to a subscriber. The services (typically identified by a service identification or SID) include telephony, television broadcasts, internet communications (e.g., WWW), facsimile, file data transfer, electronic mailing services (email), video conferencing, live or time-delayed feeds (such as, speeches, debates, presentations, news reports, sporting events, concerts, etc.), and the like. Hence, the data exchanged between CMTS 102 and cable modems 104 includes text, video, audio, voice, graphics, other media or a combination thereof (i.e., multimedia).

[0037] Cable modem 104 includes an output queue 112, CM scheduler 114 and burst multiplexer 114. Each service provided by cable modem 104 is mapped to one or more priority queues (not shown) within output queue 112. CM scheduler 114 is responsible for deciding the order in which packets are sent, and for controlling and balancing the request/grant loop process for all services. At the appropriate time, CM scheduler 114 directs packets to be sent to burst multiplexer 116, where the packets are multiplexed into a burst.

[0038] MAP messages from CMTS 102 are recovered with other messages by a demultiplexer 142. Grants in the MAP messages containing the slot structure and grants for the requesting cable modem 104 are separated and sent to CM scheduler 114 to control the allocation of packets to the granted bandwidth regions. The C/NC signal transmitted downstream and the CRA parameter derived from a control message are sent downstream and used by the downstream CRA 144 to adopt the parameters. CRA 144 sends the count of contention regions to CM scheduler 114. These counts corresponds to the number of priority CMSs the cable model 104 must wait before it can transmit a request in a contention bandwidth region of the same priority.

[0039] Input queue 150 stores data received from the upstream until it is ready to be processed by the services. Upper layer 148 receives packets from the services and forwards them to output queue 112.

II. Cable Modem Scheduling

[0040] Referring to FIG. 2, flowchart 200 represents the general operational flow of an embodiment of the present invention. More specifically, flowchart 200 shows an example of a control flow for scheduling data transmissions from cable modem 104 over communication infrastructure 110.

[0041] Referring to FIG. 2, the control flow of flowchart 200 begins at step 201 and passes immediately to step 204. At step 204, output queue 112 receives a data packet from a service (e.g., telephony, cable, and the like) and stores the packet within the appropriate priority queue (not shown). Output queue 112 notifies CM scheduler 114 of its queue state on a periodically scheduled basis. In an embodiment, output queue 112 transmits its queue state each time it is modified.

[0042] At step 208, upon notification from output queue 112, CM scheduler 114 decides whether to send a request message for bandwidth to CMTS 102, based on internal policies or rules. CMTS 102 prepares a grant specification to allocate

bandwidth according to the size specified in the bandwidth request. The grant specification is transmitted to the requesting cable modem 104.

[0043] At step 212, CM scheduler 114 receives the grant specification from CMTS 102, and at step 216, CM scheduler 114 evaluates the needs of the service(s) being provided by cable modem 104. In an embodiment, CM scheduler 114 evaluates the needs by considering the current queue state of the priority queues for each service. The current queue state can be evaluated by measuring the quantity of packets, bandwidth size, byte size, or the like. In an embodiment, CM scheduler 114 evaluates the needs of the service(s) by balancing throughput requirements versus latency. For example, it may become necessary to interrupt or fragment a transmission of text data to allow a voice transmission since voice communication require a lower tolerance for delay.

[0044] At step 220, CM scheduler 114 determines which packets to send, based on the needs assessment performed at step 212. Therefore, CM scheduler 114 is not required to use a grant in the same order that the corresponding requests were sent. CM scheduler 114 functions as a bandwidth manager that decides how to use the received grants according to the current needs of an active service. Since the needs of a service can change from the time of requesting a grant, CM scheduler 114 is programmable to assign a particular granted region (or portion thereof) to a different service than the one specified in the grant specification.

[0045] In an embodiment, CM scheduler 114 is priority based and will empty the priority queue for a higher-priority service before drawing data from the priority queue for a lower-priority service. For example, a higher-priority service (such as, telephony) and a lower-priority service (such a, web browsing) can request bandwidth from CMTS 102. The flexible use of grants provided by cable modem 104 allows the higher-priority service (i.e., telephony) to borrow a first arriving lower-priority grant if the grant arrives earlier than its own grant. The lower-priority service (i.e., web browsing) would be permitted to utilize the bandwidth

granted to the higher-priority, unless another higher-priority service is judged to require the grant.

[0046] In an embodiment, each service is registered as being a borrower, lender, both or none. As a borrower, the service is permitted to transmit data in a slot granted to another service. As a lender, the service is permitted to allow another service to transmit data over a slot granted to the lender. If registered as none, the service is not permitted to lend or borrow grants. Finally, as both, the service operates as a lender and borrower.

[0047] Thus, cable modem 104 is a flexible modem. In other words, CM scheduler 114 is configurable to overwrite a centralized CMTS scheduling decision in a seamless manner, such that the overwriting is virtually undetectable by CMTS 102 or the subscriber receiving the service. As described above, conventional systems (e.g., a DOCSIS-compliant system) must follow the instructions given in a grant. However, CM scheduler 114 decides which priority queues to transmit at the time the data is sent, instead of maintaining the decisions made at the requesting phase. Accordingly, CM scheduler 114 is programmable to change decisions at any time ranging from when CM scheduler 114 first sends a request until it transmits the actual information.

[0048] Nonetheless, the grant slots must be scheduled such that CMTS 102 detects a matching between the amount of data requested from a service and the actual amount of data the service is transmitting. In other words, CM scheduler 104 must balance the request/grant loop for each individual service. Additionally, each service must request bandwidth even if it has nothing to send in queue 112 but has used the grants apportioned to other services. As a result, the present invention guarantees that a grant for the other services will be available at some future point in time. Moreover, an advantage of the present invention is that cable modem 104 can reduce the latency of higher-priority services, which manifests a substantial improvement in quality of service. Another advantage of the present invention is that cable modem 104 can operate with more services than the ones

that CMTS 102 may have recognized at a given point in time. This is a transient advantage because, as discussed, cable modem 104 permits a borrowing service to use the grant of a lending service, only if a bandwidth request will be transmitted for the borrowing service.

[0049] Referring again to FIG. 2, after CM scheduler 114 has selected the packets to be transmitted, the control flow passes to step 224, where burst multiplexer 116 formats the data packets and transmits a burst to CMTS 102. After the burst has been transmitted, the control flow ends as indicated by step 295.

III. Piggybacking Bandwidth Requests

[0050] Referring back to FIG. 2, at step 208, CM scheduler 114 transmits bandwidth requests in either contention or piggyback mode. As discussed above, any request transmitted in contention mode bears the risk of being corrupted or loss due to collision.

[0051] Piggyback mode can be used to reduce the load of requests in the contention channel. In an embodiment, CM scheduler 114 prepares a piggyback request message that is formatted to have the highest priority for transmissions. Convention piggybacking requests are included as part of an extended header of another message. However, the piggyback request messages of the present invention are separate messages that are transmitted in a contention channel or a reservation channel without making a previous reservation. As used herein, traditional piggybacking is referred to as being piggyback requests in extended headers and independent piggybacking is referred to as being piggyback requests that are sent as separate messages.

[0052] Since concatenating messages typically do not introduce any additional overhead, the efficiency of independent piggybacking is comparable to the efficiency gained by using traditional piggybacking for systems with smaller header sizes. For example, an implicit convention for voice packets is the

specification of very small headers. An advantage of independent piggybacking is that the piggyback request message can be easily sent anywhere in a burst (i.e., between packets) without imposing any processing delays on CMTS 102, in particular in cases where headers cannot be easily extended.

[0053] As discussed with reference to step 220 as shown in FIG. 2, CM scheduler 114 has the flexibility to use grant slots to optimize throughput and reduce latency. As such, a piggyback request message can be transmitted at anytime upon receipt of a grant specification or in contention mini-slots. The piggyback request message can be inserted in a burst of voice packets or other data packets (i.e., text, graphics). In an embodiment, CM scheduler 114 uses cross-piggybacking to combine a piggyback request message for one service (i.e., primary piggyback) with a piggyback request messages from one or more other services (i.e., secondary piggyback(s)). Although secondary piggybacks are generally requests from other services, secondary piggybacks can also be from the same service. For example, a service may request more bandwidth than the maximum request size imposed by system 100. In addition, CM scheduler 114 may decide to send another request before it receives a grant for a previously transmitted request. Therefore, a cable modem 104 can have more than one piggyback request outstanding in CMTS 102, at any given time.

[0054] FIGs. 4a-4b illustrate an operational flow for requesting and granting bandwidth according to an embodiment of the present invention. Referring to FIG. 4a, the operational states of cable 104 are shown as being in either an open state 402 or closed state 404. Open state 402 indicates that cable modem 104 has one or more requests outstanding. Closed state 404 indicates that cable modem 104 has no outstanding requests.

[0055] FIG. 4b shows the operational flow of the various queue states for cable modem 104, according to an embodiment of the present invention. More specifically, when cable modem 104 is operating in open state 402, cable modem 104 can operate in one or more of four queues states, namely requesting state 406,

waiting state 408, acknowledge state 409 and close state 410. Requesting state 406 indicates that CM scheduler 114 is requesting bandwidth from CMTS 102. Once the request has been transmitted, cable modem 104 enters waiting state 408 until feedback is received from CMTS 102. Acknowledge state 409 indicates that CMTS 102 has received the request or cable modem 104 has received a corresponding grant. If cable modem 104 remains in acknowledge state 409 or waiting state 408 beyond a predetermined time, cable modem 104 generate another request (i.e., re-enter request state 406). In other words, if cable modem 104 does not receive an acknowledgment message or grant message within a predetermined time frame, cable modem 104 will generate another request.

[0056] After all requests have been granted or there are no other pending or outstanding requests, cable modem 104 enters a close state 410. Close state 410 indicates that either output queue 112 has not recently signaled CM scheduler 114 for additional bandwidth or all requests have been granted or are no longer required.

IV. Conclusion

[0057] FIG. 1 is a conceptual illustration of system 100 that allows an easy explanation of the present invention. That is, one or more of the blocks can be performed by the same piece of hardware or module of software. It should also be understood that embodiments of the present invention can be implemented in hardware, software, or a combination thereof. In such an embodiment, the various components and steps would be implemented in hardware and/or software to perform the functions of the present invention.

[0058] Additionally, the present invention (e.g., system 100 or any part thereof) can be implemented in one or more computer systems or other processing systems. In fact, in one embodiment, the invention is directed toward one or more computer systems capable of carrying out the functionality described herein.

[0062] In alternative embodiments, secondary memory 310 can include other similar means for allowing computer programs or other instructions to be loaded into computer system 300. Such means can include, for example, a removable storage unit 322 and an interface 320. Examples of such can include a program cartridge and cartridge interface (such as that found in video game devices), a removable memory chip (such as an EPROM, or PROM) and associated socket, and other removable storage units 322 and interfaces 320 which allow software

and data to be transferred from the removable storage unit 322 to computer system 300.

[0063] Computer system 300 can also include a communications interface 324. Communications interface 324 allows software and data to be transferred between computer system 300 and external devices. Examples of communications interface 324 can include a modem, a network interface (such as an Ethernet card), a communications port, a PCMCIA slot and card, etc. Software and data transferred via communications interface 324 are in the form of signals 328 which can be electronic, electromagnetic, optical or other signals capable of being received by communications interface 324. These signals 328 are provided to communications interface 324 via a communications path (i.e., channel) 326. This channel 326 carries signals 328 and can be implemented using wire or cable, fiber optics, a phone line, a cellular phone link, an RF link and other communications channels.

[0064] In this document, the terms “computer program medium” and “computer usable medium” are used to generally refer to media such as removable storage drive 314, a hard disk installed in hard disk drive 312, and signals 328. These computer program products are means for providing software to computer system 300. The invention is directed to such computer program products.

[0065] Computer programs (also called computer control logic) are stored in main memory 308 and/or secondary memory 310. Computer programs can also be received via communications interface 324. Such computer programs, when executed, enable the computer system 300 to perform the features of the present invention as discussed herein. In particular, the computer programs, when executed, enable the processor 304 to perform the features of the present invention. Accordingly, such computer programs represent controllers of the computer system 300.

[0066] In an embodiment where the invention is implemented using software, the software can be stored in a computer program product and loaded into computer

system 300 using removable storage drive 314, hard drive 312 or communications interface 324. The control logic (software), when executed by the processor 304, causes the processor 304 to perform the functions of the invention as described herein.

[0067] In another embodiment, the invention is implemented primarily in hardware using, for example, hardware components such as application specific integrated circuits (ASICs). Implementation of the hardware state machine so as to perform the functions described herein will be apparent to persons skilled in the relevant art(s).

[0068] In yet another embodiment, the invention is implemented using a combination of both hardware and software.

[0069] While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example, and not limitation. It will be apparent to persons skilled in the relevant art(s) that various changes in form and detail can be made therein without departing from the spirit and scope of the invention. Moreover, it should be understood that the method and system of the present invention should not be limited to transmissions between cable modems and headends. The present invention can be implemented in any multi-nodal communications environment governed by a centralized node. The nodes can include communication gateways, switches, routers, Internet access facilities, servers, personal computers, enhanced telephones, personal digital assistants (PDA), televisions, set-top boxes or the like. Thus, the present invention should not be limited by any of the above described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.